

CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY**DEVANG PATEL INSTITUTE OF ADVANCE TECHNOLOGY & RESEARCH**

Department of Computer Engineering

Subject Name: Java Programming**Semester: III****Subject Code: CE251****Academic year: 2023-24****PART-I Data Types, Variables, String, Control Statements, Operators, Arrays****Practical – 1**

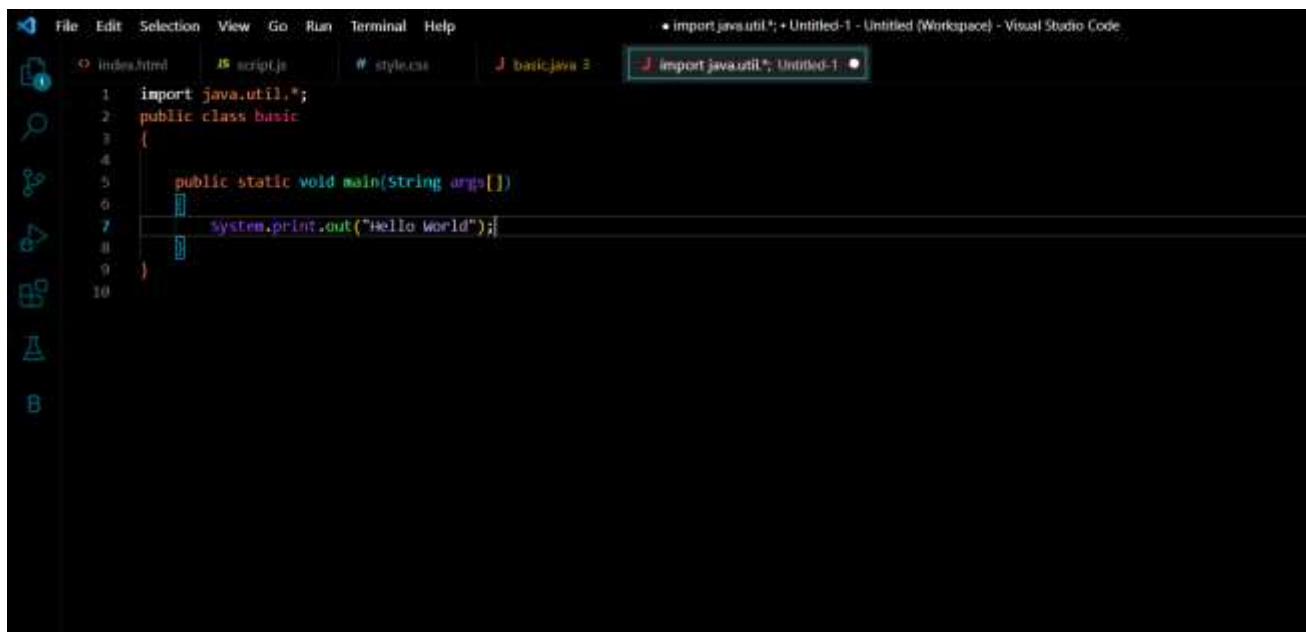
Introduction to Object Oriented Concepts, comparison of Java with other object-oriented programming languages. Introduction to JDK, JRE, JVM, Javadoc, command line argument. Introduction to Eclipse or NetBeans IDE, or Blue-J and Console Programming.

- Object-oriented programming is a core of Java Programming, which is used for designing a program using classes and objects.
- **C++ vs JAVA**

C++	JAVA
C++ is platform dependent.	Java is platform-independent.
C++ supports internal and external pointers.	Java does not have an external pointer.
C++ is compiled and run using the compiler which converts the source code to machine code.	Java uses a compiler and interpreter. It compiles a Java file to a bytecode file.
C++ support structure and union.	Java does not support structure and union

- **JDK – Java Development Kit**
- **JRE – Java Runtime Environment**
- **JVM – Java Virtual Machine**

```
C:\Users\Harsh Bhagchandani>java -version
java version "20.0.1" 2023-04-18
Java(TM) SE Runtime Environment (build 20.0.1+9-29)
Java HotSpot(TM) 64-Bit Server VM (build 20.0.1+9-29, mixed mode, sharing)
```



Practical – 2

Install Java and show step by step installation.

Follow these steps to setup and install Java on Windows:

1. Using your preferred web browser, navigate to the [Oracle Java Downloads page](#).
2. On the *Downloads* page, click the **x64 Installer** download link under the **Windows** category. At the time of writing this article, Java version 17 is the latest long-term support Java version.

Linux	macOS	Windows
Product/file description		
		File size
		Download
x64 Compressed Archive		170.66 MB
		https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.zip (sha256 [icon])
x64 Installer		152 MB
		https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.exe (sha256 [icon])
x64 MSI Installer		150.89 MB
		https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.msi (sha256 [icon])

Step 3: Run the Downloaded File

Double-click the **downloaded file** to start the installation.

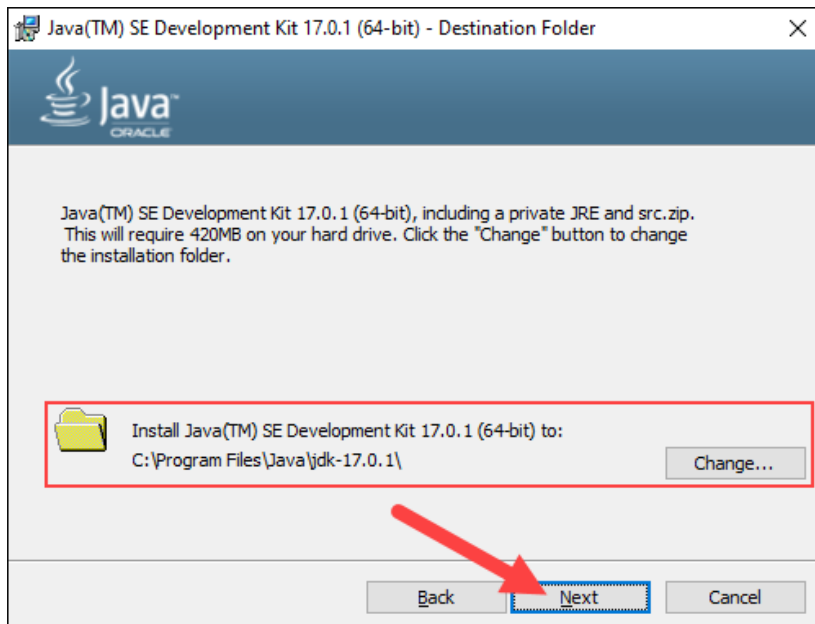
Step 4: Configure the Installation Wizard

After running the installation file, the installation wizard welcome screen appears.

1. Click **Next** to proceed to the next step.



5. Choose the destination folder for the Java installation files or stick to the default path. Click **Next** to proceed.



6. Wait for the wizard to finish the installation process until the *Successfully Installed* message appears. Click **Close** to exit the wizard.

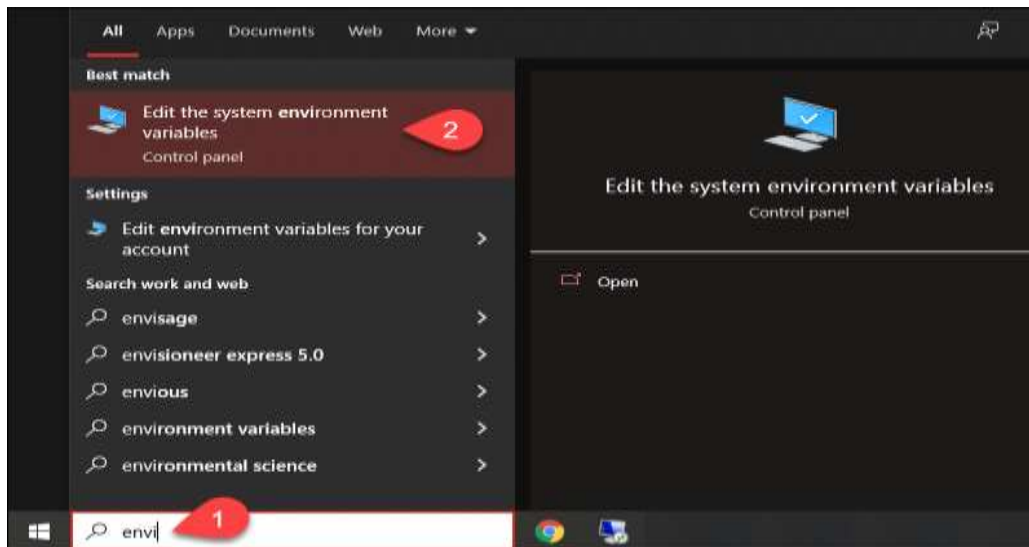


Set Environmental Variables in Java

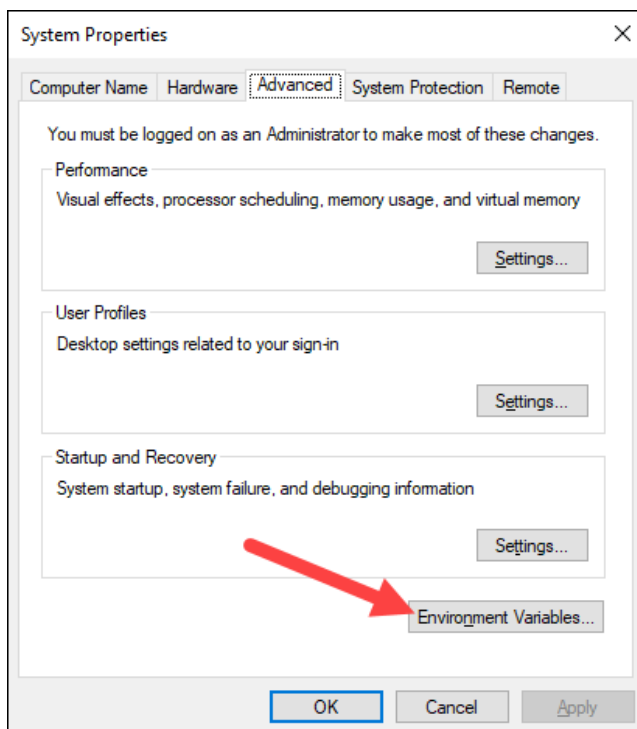
Set Java [environment variables](#) to enable program compiling from any directory. To do so, follow the steps below:

Step 7: Add Java to System Variables

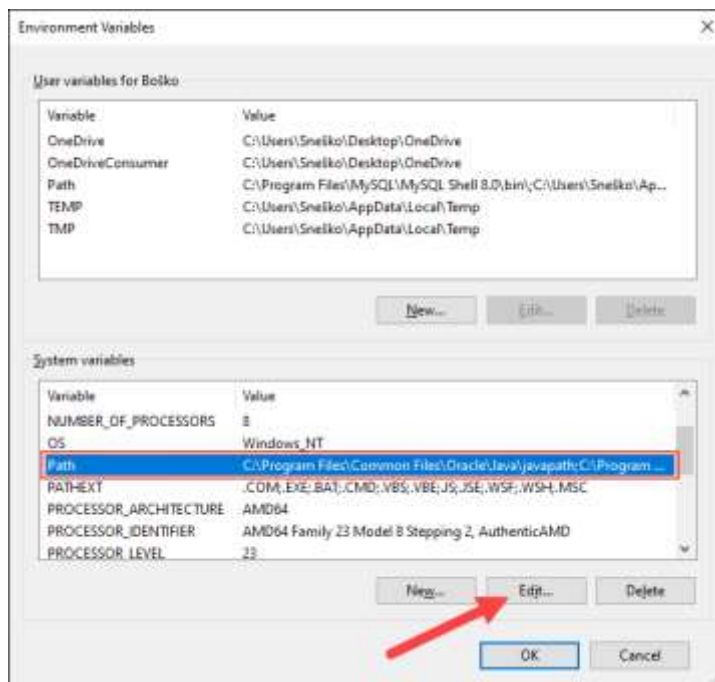
1. Open the **Start** menu and search for *environment variables*.
2. Select the **Edit the system environment variables** result.



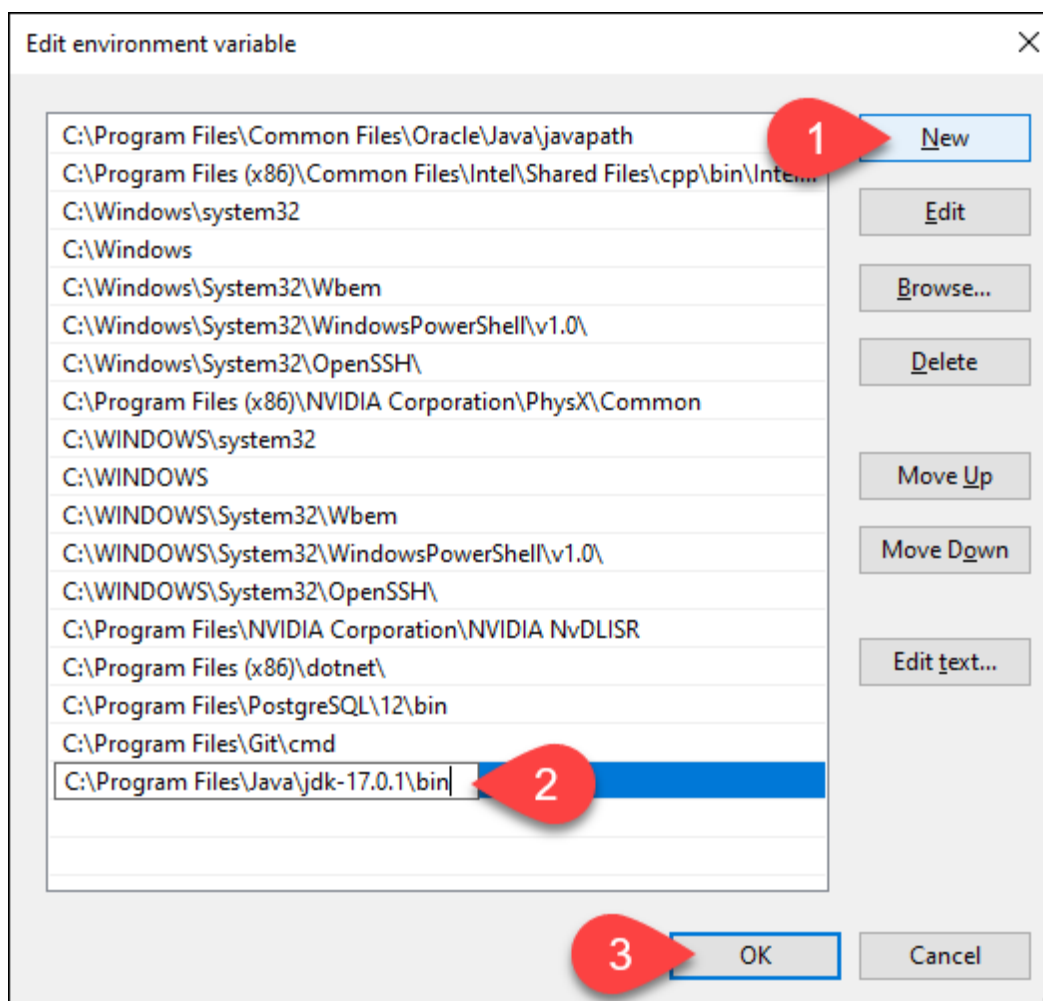
8. In the *System Properties* window, under the *Advanced* tab, click **Environment Variables...**



9. Under the *System variables* category, select the **Path** variable and click **Edit**:



10. Click the **New** button and enter the path to the Java bin directory:



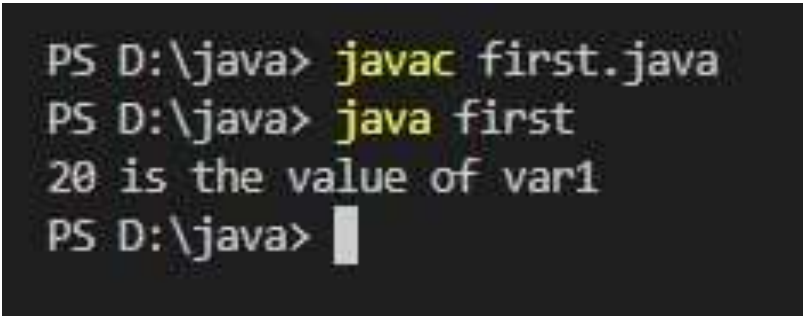
Practical-3

Write a program that declares one integer variable called var1. Give value 20 to this variable and then, using one println() statement, display the value on the screen like this: “20 is the value of var1.”

Program Code:

```
import java.util.*;
public class first
{
    public static void main(String args[])
    {
        int var1=20;
        System.out.println(var1+" is the value of var1");
    }
}
```

OUTPUT



```
PS D:\java> javac first.java
PS D:\java> java first
20 is the value of var1
PS D:\java> █
```

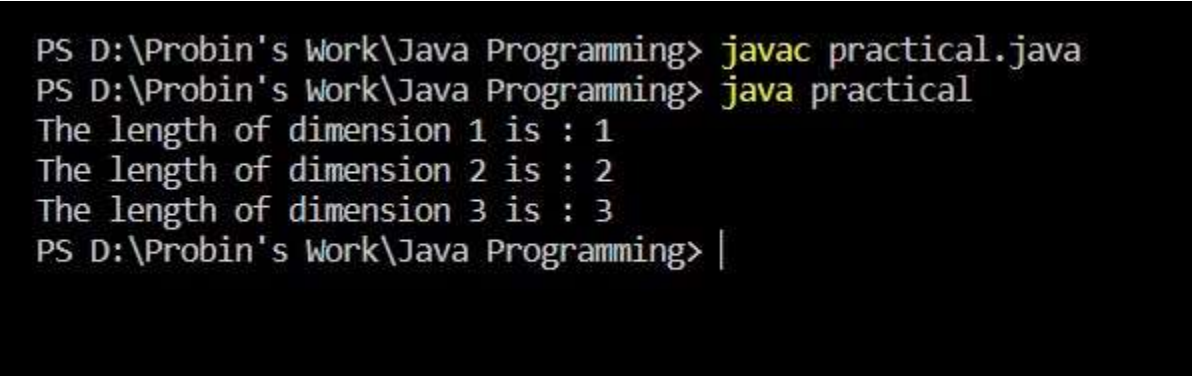
Practical-4

Write an application that creates a two-dimension array with int values. The first, second and third elements should be arrays with one, two and three numbers respectively. Display the length of each dimension.

PROGRAM CODE:

```
public class practical
{
    public static void main(String[] args)
    {
        int[][] arr = {{1}, {2,3}, {4,5,6}};
        int i;
        for(i=0; i<arr.length ; i++)
        {
            System.out.println("The length of dimension " + (i+1)+" is : " + arr[i].length);
        }
    }
}
```

OUTPUT



```
PS D:\Probin's Work\Java Programming> javac practical.java
PS D:\Probin's Work\Java Programming> java practical
The length of dimension 1 is : 1
The length of dimension 2 is : 2
The length of dimension 3 is : 3
PS D:\Probin's Work\Java Programming> |
```


Practical-5

An electric appliance shop assigns code 1 to the motor, 2 to the fan, 3 to the tube and 4 to the wires. All other items have code 5 or more. While selling the goods, a sales tax of 8% to the motor, 12% to fan, 5% to the tube light, 7.5% to wires and 3% for all other items is charged. A list containing the product code and price in two different arrays. Write a Java program using switch statement to prepare the bill.

Program Code:

```
import java.util.*;
public class first
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int code[]={ 1,2,3,4,5};
        int price[]={ 100,200,300,400,500};
        double bill=0;
        System.out.println("Enter the code 1 for motor , 2 for fan , 3 for tube , 4 for wires");
        int usercode=sc.nextInt();
        switch(usercode)
        {
            case 1:
                bill=(double)price[usercode-1] + (0.08*price[usercode-1]);
                System.out.println("Bill for motor is: "+bill);
                break;

            case 2:
                bill=(double)price[usercode-1] + (0.12*price[usercode-1]);
                System.out.println("Bill for fan is: "+bill);
                break;

            case 3:
                bill=(double)price[usercode-1] + (0.05*price[usercode-1]);
                System.out.println("Bill for tube is: "+bill);
                break;

            case 4:
                bill=(double)price[usercode-1] + (0.075*price[usercode-1]);
                System.out.println("Bill for wire is: "+bill);
                break;

            default:
                bill=(double)price[usercode-1] + (0.03*price[usercode-1]);
                System.out.println("Bill for other is: "+bill);
                break;
        }
    }
}
```

```
}  
  
}
```

OUTPUT

```
HELLO WORLD  
Enter the code 1 for motor , 2 for fan , 3 for tube , 4 for wires  
1  
PS D:\java> java first  
HELLO WORLD  
Enter the code 1 for motor , 2 for fan , 3 for tube , 4 for wires  
2  
Bill for fan is: 224.0  
PS D:\java> java first  
HELLO WORLD  
Enter the code 1 for motor , 2 for fan , 3 for tube , 4 for wires  
3  
Bill for tube is: 315.0  
PS D:\java> java first  
HELLO WORLD  
Enter the code 1 for motor , 2 for fan , 3 for tube , 4 for wires  
4  
Bill for wire is: 430.0  
PS D:\java> java first  
HELLO WORLD  
Enter the code 1 for motor , 2 for fan , 3 for tube , 4 for wires  
5  
Bill for other is: 515.0
```

Practical-6

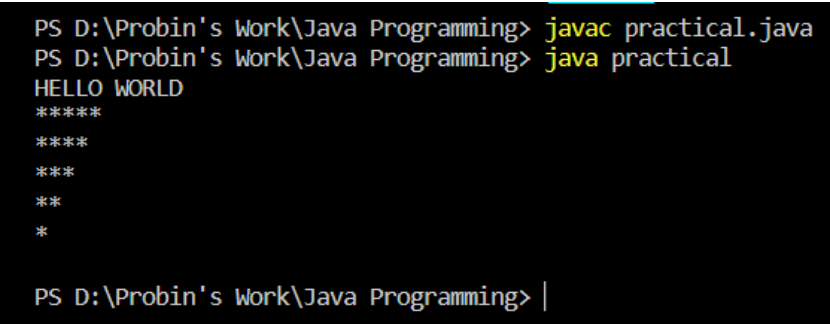
Write a program to show output like:

```
*****  
*****  
***  
**  
*
```

Program Code

```
import java.util.*;  
public class practical  
{  
    public static void main(String args[])  
    {  
        Scanner sc=new Scanner(System.in);  
        System.out.println("HELLO WORLD");  
        int i,j,k,n=5;  
        for(i=n; i>=1 ; i--)  
        {  
            for(j=1; j<=i; j++)  
            {  
                System.out.print("*");  
            }  
            System.out.println();  
        }  
        System.out.println();  
    }  
}
```

OUTPUT



```
PS D:\Probin's Work\Java Programming> javac practical.java  
PS D:\Probin's Work\Java Programming> java practical  
HELLO WORLD  
*****  
*****  
***  
**  
*  
  
PS D:\Probin's Work\Java Programming> |
```

Staff Signature:

Grade:

Remarks by the Staff:

PART-II String

Practical – 7

Given a string and a non-negative int n, we'll say that the front of the string is the first 3 chars, or whatever is there if the string is less than length 3. Return n copies of the front; `front_times('Chocolate', 2) → 'ChoCho'` `front_times('Chocolate', 3) → 'ChoChoCho'` `front_times('Abc', 3) → 'AbcAbcAbc'`.

Program Code:

```
import java.util.*;
public class nm
{
    public static String front_times(String a , int x)
    {
        String answer="";
        int flen=3;
        if(flen>a.length())
        {
            flen=a.length();
        }
        for(int i=0 ; i<x ; i++)
        {
            answer=answer+ a.substring(0,flen);
        }
        return answer;
    }
    public static void main(String args[])
    {
        System.out.println("22DCE006");
        Scanner sc=new Scanner(System.in);
        String a;
        a=sc.nextLine();
        String y=front_times(a , 3);
        System.out.println("Final Answer is: "+y);
    }
}
```

OUTPUT

```
PS C:\Users\Administrator> javac nm.java
PS C:\Users\Administrator> java nm
22DCE006
Chocolate
Final Answer is: ChoChoCho
PS C:\Users\Administrator> java nm
22DCE006
ab
Final Answer is: ababab
PS C:\Users\Administrator> |
```

Practical – 8

Given an array of ints, return the number of 9's in the array.

`array_count9([1, 2, 9]) → 1`

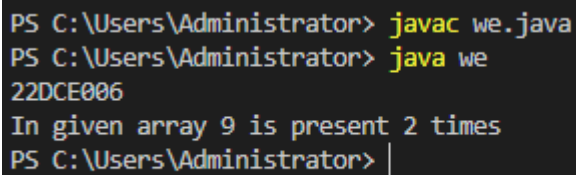
`array_count9([1, 9, 9]) → 2`

`array_count9([1, 9, 9, 3, 9]) → 3`

Program Code:

```
import java.util.*;
public class we
{
    public static int Count_Fun(int arr[])
    {
        int counter=0;
        for(int i=0 ; i<arr.length ; i++)
        {
            if(arr[i]==9)
            {
                counter++;
            }
        }
        return counter;
    }
    public static void main(String args[])
    {
        System.out.println("22DCE006");
        Scanner sc=new Scanner(System.in);
        int arr[]={1,9,2,3,9};
        int y=Count_Fun(arr);
        System.out.println("In given array 9 is present "+y+" times");
    }
}
```

OUTPUT



```
PS C:\Users\Administrator> javac we.java
PS C:\Users\Administrator> java we
22DCE006
In given array 9 is present 2 times
PS C:\Users\Administrator> |
```

Practical – 9

Given a string, return a string where for every char in the original, there are two chars.

`double_char('The') → 'TThhee'`

`double_char('AAbb') → 'AAAAbbbb'`

`double_char('Hi-There') → 'HHii--TThheerree'`

Program Code:

```
import java.util.*;
public class hi
{
    public static String front_times(String a )
    {
        String w="";
        for(int j=0 ; j<a.length() ; j++)
        {
            for(int i=0 ; i<2; i++)
            {
                w=w+ a.substring(j, j+1);
            }
        }
        return w;
    }
    public static void main(String args[])
    {
        System.out.println("22DCE006");
        Scanner sc=new Scanner(System.in);
        String a="ABCDEF";
        String y=front_times(a);
        System.out.println("Final Answer is: "+y);
    }
}
```

OUTPUT:

```
PS C:\Users\Administrator> javac hi.java
PS C:\Users\Administrator> java hi
22DCE006
Final Answer is: AABCCDDEEFF
PS C:\Users\Administrator> |
```

Practical – 10

Perform following functionalities of the string:

- **Find Length of the String**
- **Lowercase of the String**
- **Uppercase of the String**
- **Reverse String**
- **Sort the string**

Program Code:

```
import java.util.*;
public class rough
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("22DCE006");
        String str="depstar";
        int len=str.length();
        System.out.println("The length of given string is: "+len);
        String new1=str.toLowerCase();
        System.out.println("The lowercase of given string is: "+new1);
        String new2=str.toUpperCase();
        System.out.println("The uppercase of given string is: "+new2);

        String reversedStr = "";
        for (int i = 0; i < str.length(); i++)
        {
            reversedStr = str.charAt(i) + reversedStr;
        }
        System.out.println("The reverse of given string is: "+reversedStr);
        System.out.println("The sorted string is as follows: ");
        char arr[] = str.toCharArray() , char temp;
        int i = 0;
        while (i < arr.length) {
            int j = i + 1;
            while (j < arr.length) {
                if (arr[j] < arr[i]) {
                    temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
                j += 1;
            }
            i += 1;
        }
        System.out.println(arr);
    }
}
```

OUTPUT:

```
PS D:\Probin's Work\Java Programming> javac rough.java
PS D:\Probin's Work\Java Programming> java rough
22DCE006
The length of given string is: 7
The lowercase of given string is: depstar
The uppcase of given string is: DEPSTAR
The reverse of given string is: ratsped
The sorted string is as follows:
adeprst
PS D:\Probin's Work\Java Programming> |
```


Practical – 11

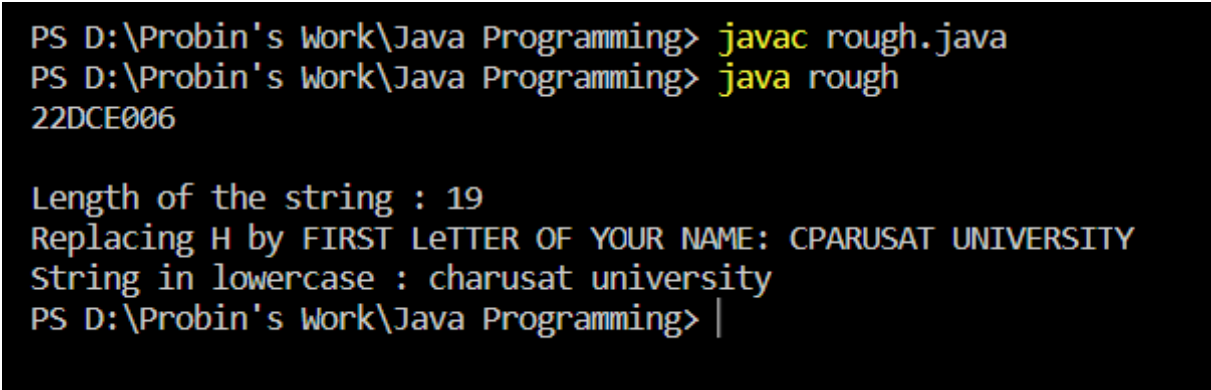
Perform following Functionalities of the string: “CHARUSAT UNIVERSITY”

- Find length
- Replace ‘H’ by ‘FIRST LETTER OF YOUR NAME’
- Convert all character in lowercase

Program Code:

```
import java.util.*;
public class rough
{
    public static void main(String[] args)
    {
        System.out.println("22DCE006");
        String str = "CHARUSAT UNIVERSITY";
        System.out.print("\nLength of the string : " + str.length());
        System.out.print("\nReplacing H by FIRST LeTTER OF YOUR NAME: " + str.replace('H','P'));
        System.out.print("\nString in lowercase : " + str.toLowerCase());
    }
}
```

OUTPUT:



```
PS D:\Probin's Work\Java Programming> javac rough.java
PS D:\Probin's Work\Java Programming> java rough
22DCE006

Length of the string : 19
Replacing H by FIRST LeTTER OF YOUR NAME: CPARUSAT UNIVERSITY
String in lowercase : charusat university
PS D:\Probin's Work\Java Programming> |
```

Staff Signature:

Grade:

Remarks by the Staff:

PART- III Object Oriented Programming: Classes, Methods, Constructors

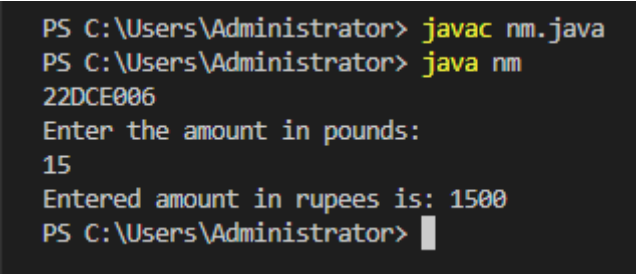
Practical – 12

Write a java program for converting Pound into Rupees. (Accept Pounds from command line argument and using scanner class also and take 1 Pound = 100 Rupees.)

Program Code:

```
import java.util.*;
public class nm{
    public static void main(String args[]){
        System.out.println("22DCE006");
        Scanner sc=new Scanner(System.in);
        int rupee,pound;
        System.out.println("Enter the amount in pounds: ");
        pound=sc.nextInt();
        int ans=pound*100;
        System.out.println("Entered amount in rupees is: "+ans);
    }
}
```

OUTPUT:



```
PS C:\Users\Administrator> javac nm.java
PS C:\Users\Administrator> java nm
22DCE006
Enter the amount in pounds:
15
Entered amount in rupees is: 1500
PS C:\Users\Administrator> █
```

Practical – 13

Create a class called **Employee** that includes three pieces of information as instance variables—a first name (type **String**), a last name (type **String**) and a monthly salary (double). Your class should have a constructor that initializes the three instance variables. Provide a set and a get method for each instance variable. If the monthly salary is not positive, set it to 0.0. Write a test application named **EmployeeTest** that demonstrates class **Employee**'s capabilities. Create two **Employee** objects and display each object's yearly salary. Then give each **Employee** a 10% raise and display each **Employee**'s yearly salary again.

Program Code:

```
import java.util.*;
class Employee
{
    Scanner sc=new Scanner(System.in);
    String first_name,last_name;
    double salary;
    public void get()
    {
        System.out.println("Enter first name: ");
        first_name=sc.nextLine();

        System.out.println("Enter last name: ");
        last_name=sc.nextLine();

        System.out.println("Enter salary: ");
        salary=sc.nextDouble();
        if(salary<0)
        {
            salary=0.0;
        }
    }
    public void put()
    {
        System.out.println("The first name: "+first_name);
        System.out.println("The last name: "+last_name);
        System.out.println("The salary: "+salary);
    }
    public void raise(float per)
    {
        double raise=(per*salary)/(100);
        salary=salary+raise;
        System.out.println("The salary with yearly raise is: "+salary);
    }
}

public class op
{
    public static void main(String args[])
    {
        System.out.println("22DCE006");
    }
}
```

```
Employee e1=new Employee();
e1.get();
e1.put();
e1.raise(10);
Employee e2=new Employee();
e2.get();
e2.put();
e2.raise(20);
}
}
```

OUTPUT:

```
PS C:\Users\Administrator> javac op.java
PS C:\Users\Administrator> java op
22DCE006
Enter first name:
ramesh
Enter last name:
patel
Enter salary:
1000
The first name: ramesh
The last name: patel
The salary: 1000.0
The salary with yearly raise is: 1100.0
Enter first name:
suresh
Enter last name:
shah
Enter salary:
5000
The first name: suresh
The last name: shah
The salary: 5000.0
The salary with yearly raise is: 6000.0
PS C:\Users\Administrator> |
```

Practical – 14

Create a class called **Date** that includes three pieces of information as instance variables—a month (type int), a day (type int) and a year (type int). Your class should have a constructor that initializes the three instance variables and assumes that the values provided are correct. Provide a set and a get method for each instance variable. Provide a method **displayDate** that displays the month, day and year separated by forward slashes (/). Write a test application named **DateTest** that demonstrates class **Date**'s capabilities.

Program Code:

```
import java.util.*;
class date
{
    Scanner sc=new Scanner(System.in);

    int year,month,day;
    public void get()
    {
        System.out.println("Enter day: ");
        day=sc.nextInt();

        System.out.println("Enter month: ");
        month=sc.nextInt();

        System.out.println("Enter year: ");
        year=sc.nextInt();
    }
    public void put()
    {
        System.out.println("The day: "+day);
        System.out.println("The year: "+year);
        System.out.println("The year: "+year);
    }
    public void display()
    {
        System.out.println("The date with format is: "+day+"/"+month+"/"+year);
    }
}
public class vb
{
    public static void main(String args[])
    {
        System.out.println("22DCE006");
        date e1=new date();
        e1.get();
        e1.put();
        e1.display();
    }
}
```

```
}  
}
```

OUTPUT:

```
PS C:\Users\Administrator> javac vb.java  
PS C:\Users\Administrator> java vb  
22DCE006  
Enter day:  
15  
Enter month:  
11  
Enter year:  
2004  
The day: 15  
The year: 2004  
The year: 2004  
The date with format is: 15/11/2004  
PS C:\Users\Administrator> |
```

Practical – 15

Write a program to print the area of a rectangle by creating a class named 'Area' taking the values of its length and breadth as parameters of its constructor and having a method named 'returnArea' which returns the area of the rectangle. Length and breadth of rectangle are entered through keyboard.

Program Code:

```
import java.util.*;
class area
{
    Scanner sc=new Scanner(System.in);

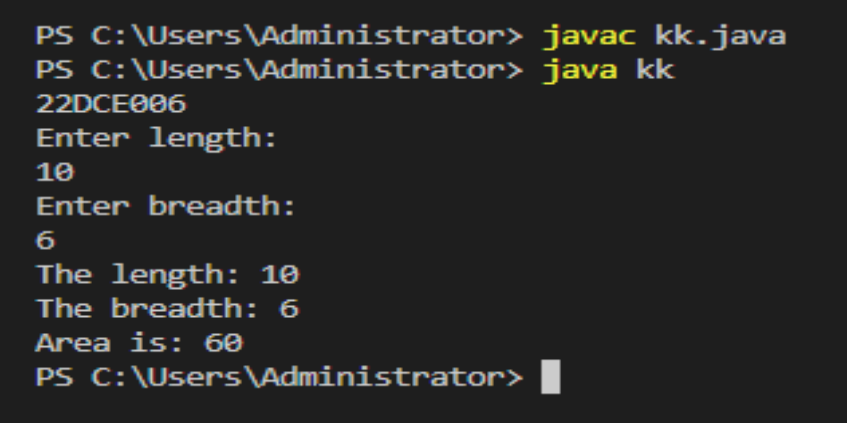
    int length,breadth;
    public void get()
    {
        System.out.println("Enter length: ");
        length=sc.nextInt();

        System.out.println("Enter breadth: ");
        breadth=sc.nextInt();

    }
    public void put()
    {
        System.out.println("The length: "+length);
        System.out.println("The breadth: "+breadth);

    }
    area(int l , int b)
    {
        l=length;
        b=breadth;
    }
    public int return_area()
    {
        return length*breadth;
    }
}
public class kk
{
    public static void main(String args[])
    {
        System.out.println("22DCE006");
        int l=0,b=0;
        area a1=new area(l,b);
        a1.get();
        a1.put();
        int y=a1.return_area();
    }
}
```

```
        System.out.println("Area is: "+y);  
    }  
}
```

OUTPUT:

```
PS C:\Users\Administrator> javac kk.java  
PS C:\Users\Administrator> java kk  
22DCE006  
Enter length:  
10  
Enter breadth:  
6  
The length: 10  
The breadth: 6  
Area is: 60  
PS C:\Users\Administrator> █
```


Practical – 16

Print the sum, difference and product of two complex numbers by creating a class named 'Complex' with separate methods for each operation whose real and imaginary parts are entered by user.

Program Code:

```
import java.util.*;
class complex
{
    Scanner sc=new Scanner(System.in);
    int real,img;

    public void get()
    {
        System.out.println("Enter real part: ");
        real=sc.nextInt();
        System.out.println("Enter imaginary part: ");
        img=sc.nextInt();
    }
    public void put()
    {
        System.out.println("The real part: "+real);
        System.out.println("The imaginary part: "+img);
        System.out.println("The number is "+real+" + "+img+"i");
    }
    public void sum (complex a , complex b)
    {
        real=a.real+b.real;
        img=a.img+b.img;
        System.out.println("The sum is "+real+" + "+img+"i");
    }
    public void sub(complex m , complex n)
    {
        real=m.real-n.real;
        img=m.img-n.img;
        System.out.println("The difference is "+real+" + "+img+"i");
    }
    public void product (complex c , complex d)
    {
        real=c.real*d.real;
        img=c.img*d.img;
        System.out.println("The product is "+real+" + "+img+"i");
    }
}
public class ty
{
}
```

```
public static void main(String args[])
{
    System.out.println("22DCE006");
    complex e1=new complex();
    e1.get();
    e1.put();

    complex e2=new complex();
    e2.get();
    e2.put();
    complex x=new complex();
    x.sum(e1,e2);
    x.sub(e1,e2);
    x.product(e1,e2);
}
}
```

OUTPUT:

```
PS C:\Users\Administrator> javac ty.java
PS C:\Users\Administrator> java ty
22DCE006
Enter real part:
1
Enter imaginary part:
2
The real part: 1
The imaginary part: 2
The number is 1 + 2i
Enter real part:
4
Enter imaginary part:
5
The real part: 4
The imaginary part: 5
The number is 4 + 5i
The sum is 5 + 7i
The difference is -3 + -3i
The product is 4 + 10i
PS C:\Users\Administrator> |
```

Staff Signature:**Grade:****Remarks by the Staff:**

Practical – 17

Create a class with a method that prints "This is parent class" and its subclass with another method that prints "This is child class". Now, create an object for each of the class and call 1 - method of parent class by object of parent class 2 - method of child class by object of child class 3 - method of parent class by object of child class

Program Code:

```
class parent
{
    void parentfun()
    {
        System.out.println("This is parent class");
    }
}
class child extends parent
{
    void childfun()
    {
        System.out.println("This is child class");
    }
}

public class prac
{
    public static void main(String args[])
    {
        System.out.println("22DCE006");
        Scanner sc=new Scanner(System.in);
        parent p=new parent();
        child c=new child();
        p.parentfun();
        c.childfun();
        c.parentfun();
    }
}
```

Output:

```
PS C:\java> javac prac.java
PS C:\java> java prac
22DCE006
This is parent class
This is child class
This is parent class
PS C:\java> |
```

Practical – 18

Create a class named 'Member' having the following members: Data members

1 - Name

2 - Age

3 - Phone number

4 - Address

5 – Salary

It also has a method named 'printSalary' which prints the salary of the members. Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same.

Program Code:

```
import java.util.*;

public class o {
    public static void main(String[] args)
    {
        System.out.println(" 22DCE006 ");
        Employee E1 = new Employee();
        Manager M1 = new Manager();
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Employee Details:-");
        System.out.println("Name:-");
        E1.Name = sc.next();
        System.out.println("Age :-");
        E1.age = sc.nextDouble();
        System.out.println("Phone Number");
        E1.phone = sc.nextDouble();
        System.out.println("Address:-");
        E1.Address = sc.next();
        System.out.println("salary");
        E1.salary = sc.nextDouble();
        System.out.println("Specialization:-");
        E1.s = sc.next();
        System.out.println("-----");
        System.out.println("Enter Manager Details:-");
        System.out.println("Name:-");
```

```
M1.Name = sc.next();
System.out.println("Age :-");
M1.age = sc.nextDouble();
System.out.println("Phone Number");
M1.phone = sc.nextDouble();
System.out.println("Address:-");
M1.Address = sc.next();
System.out.println("salary");
M1.salary = sc.nextDouble();
System.out.println("Specialization:-");
M1.d= sc.next();
System.out.println("");
System.out.println("salary Of Employee is ");
E1.printsalaryary(E1.salary);
System.out.println("salary Of Manager is ");
M1.printsalaryary(M1.salary);
}
}
class Member
{
    String Name = new String();
    String Address = new String();
    double age,phone,salary;

    public void printsalaryary(double salary)
    {
        System.out.println("salaryary is " + this.salary);
    }
}
class Employee extends Member
{
    String s = new String();
}
class Manager extends Member
{
    String d= new String();
}
```

Output:

```
PS C:\java> javac o.java
PS C:\java> java o
22DCE006
Enter Employee Details:-
Name:-
raj
Age :-
18
Phone Number
123456789
Address:-
manjalpur
Salary
15000
Specialization:-
computer
-----
Enter Manager Details:-
Name:-
heer
Age :-
19
Phone Number
987654321
Address:-
tulsidham
Salary
18000
Specialization:-
hr

Salary Of Employee is
Salary is 15000.0
Salary Of Manager is
Salary is 18000.0
PS C:\java> |
```

Practical – 19

Create a class named 'Rectangle' with two data members 'length' and 'breadth' and two methods to print the area and perimeter of the rectangle respectively. Its constructor having parameters for length and breadth is used to initialize length and breadth of the rectangle. Let class 'Square' inherit the 'Rectangle' class with its constructor having a parameter for its side (suppose s) calling the constructor of its parent class as 'super(s,s)'. Print the area and perimeter of a rectangle and a square. Also use array of objects.

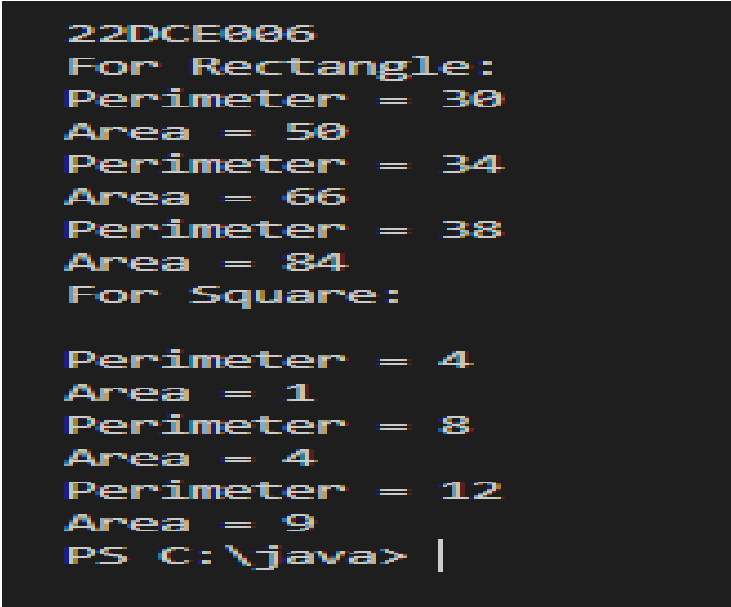
Program Code:

```
import java.util.*;
class rectangle
{
    int length;
    int width;
    public rectangle(int l , int w)
    {
        length=l;
        width=w;
    }
    public void perimeter()
    {
        System.out.println("Perimeter = "+(2*(length + width)));
    }
    public void area()
    {
        System.out.println("Area = "+(length * width));
    }
}
class square extends rectangle
{
    int side;
    public square(int s)
    {
        super(s,s);
        side=s;
    }
}

public class re
{
    public static void main(String args[])
    {
```



```
System.out.println("22DCE006");
System.out.println("For Rectangle: ");
rectangle r[]=new rectangle[3];
int x=10 , y=5;
for(int i=0 ; i<3 ; i++)
{
    r[i]=new rectangle(x,y);
    x++;
    y++;
}
for(int i=0 ; i<3 ; i++)
{
    r[i].perimeter();
    r[i].area();
}
System.out.println("For Square: ");
System.out.println();
square s[]=new square[3];
int m=1;
for(int i=0 ; i<3 ; i++)
{
    s[i]=new square(m);
    m++;
}
for(int i=0 ; i<3 ; i++)
{
    s[i].perimeter();
    s[i].area();
}
}
```

Output:

```
22DCE006
For Rectangle:
Perimeter = 38
Area = 50
Perimeter = 34
Area = 66
Perimeter = 38
Area = 84
For Square:

Perimeter = 4
Area = 1
Perimeter = 8
Area = 4
Perimeter = 12
Area = 9
PS C:\java> |
```

Practical – 20

Create a class named 'Shape' with a method to print "This is This is shape". Then create two other classes named 'Rectangle', 'Circle' inheriting the Shape class, both having a method to print "This is rectangular shape" and "This is circular shape" respectively. Create a subclass 'Square' of 'Rectangle' having a method to print "Square is a rectangle". Now call the method of 'Shape' and 'Rectangle' class by the object of 'Square' class.

Program Code:

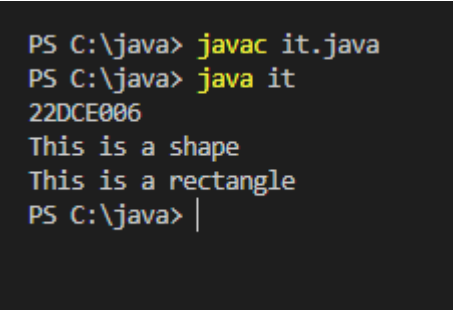
```
import java.util.*;
class shape
{
    public static void shapefun()
    {
        System.out.println("This is a shape");
    }
}
class circle extends shape
{
    public static void circlefun()
    {
        System.out.println("This is a circle");
    }
}
class rectangle extends shape
{
    public static void rectfun()
    {
        System.out.println("This is a rectangle");
    }
}

class square extends rectangle
{
    public static void squarefun()
    {
        System.out.println("Square is a rectangle");
    }
}

public class it
{
    public static void main(String args[])
    {
        shapefun();
        circlefun();
        rectfun();
        squarefun();
    }
}
```

```
{  
    System.out.println("22DCE006");  
    square s1=new square();  
    s1.shapefun();  
    s1.rectfun();  
}  
}
```

Output:



```
PS C:\java> javac it.java  
PS C:\java> java it  
22DCE006  
This is a shape  
This is a rectangle  
PS C:\java> |
```

Practical – 21

Create a class 'Degree' having a method 'getDegree' that prints 'I got a degree'. It has two subclasses namely 'Undergraduate' and 'Postgraduate' each having a method with the same name that prints 'I am an Undergraduate' and 'I am a Postgraduate' respectively. Call the method by creating an object of each of the three classes.

Program Code:

```
import java.util.*;
public class neon {
    public static void main(String[] args) {
        System.out.println("\n22DCE006");

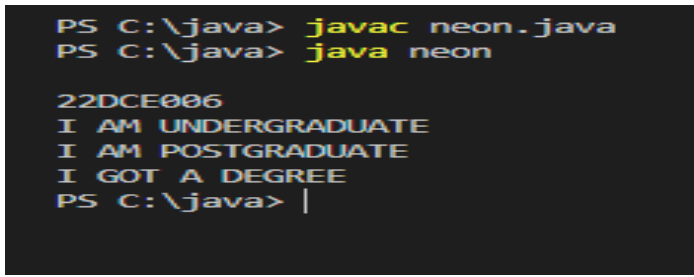
        Undergraduate u1 = new Undergraduate();
        u1.getDegree();// I got a degree
        Postgraduate p2=new Postgraduate();
        p2.getDegree();
        Degree d1 = new Degree();
        d1.getDegree();
    }
}

class Degree
{
    void getDegree(){System.out.println("I GOT A DEGREE");}
}

class Undergraduate extends Degree
{
    void getDegree(){System.out.println("I AM UNDERGRADUATE");}
}

class Postgraduate extends Degree
{
    void getDegree(){System.out.println("I AM POSTGRADUATE");}
}
```

Output:



```
PS C:\java> javac neon.java
PS C:\java> java neon


22DCE006
I AM UNDERGRADUATE
I AM POSTGRADUATE
I GOT A DEGREE
PS C:\java> |
```

Practical 22

Write a java program that implements an interface AdvancedArithmetic which contains a method signature `int divisor_sum(int n)`. You need to write a class called `MyCalculator` which implements the interface. `divisorSum` function just takes an integer as input and return the sum of all its divisors. For example, divisors of 6 are 1, 2, 3 and 6, so `divisor_sum` should return 12. The value of `n` will be at most 1000.

PROGRAM CODE:

```
interface AdvancedArithmetic {
    int divisor_sum(int n);
}
class op implements AdvancedArithmetic {
    @Override
    public int divisor_sum(int n) {
        if (n == 1) {
            return 1;
        }
        int sum = 1;
        for (int i = 2; i <= Math.sqrt(n); i++) {
            if (n % i == 0) {
                sum += i;
                if (i != n / i) {
                    sum += n / i;
                }
            }
        }
        return sum + n;
    }
}
public class clg{
    public static void main(String[] args)
    {
        System.out.println("22DCE006");
        op calculator = new op();
        int n = 6;
        int sumOfDivisors = calculator.divisor_sum(n);
        System.out.println("The sum of divisors of " + n + " is: " + sumOfDivisors);
    }
}
```

OUTPUT:

```
PROBLEMS 54 OUTPUT DEBUG CONSOLE PORTS TERMINAL
PS D:\Probin's Work\Java Programming> javac clg.java
PS D:\Probin's Work\Java Programming> java clg
22DCE006
The sum of divisors of 6 is: 12
PS D:\Probin's Work\Java Programming> |
```

Practical 23

Assume you want to capture shapes, which can be either circles (with a radius and a color) or rectangles (with a length, width, and color). You also want to be able to create signs (to post in the campus center, for example), each of which has a shape (for the background of the sign) and the text (a String) to put on the sign. Create classes and interfaces for circles, rectangles, shapes, and signs. Write a program that illustrates the significance of interface default method.

PROGRAM CODE:

```
interface Shape {
    double area();
    String getColor();
}
class Circle implements Shape {
    private double radius;
    private String color;
    public Circle(double radius, String color) {
        this.radius = radius;
        this.color = color;
    }
    @Override
    public double area() {
        return Math.PI * radius * radius;
    }
    @Override
    public String getColor() {
        return color;
    }
}
class Rectangle implements Shape {
```

```
private double length;
private double width;
private String color;
public Rectangle(double length, double width, String color) {
    this.length = length;
    this.width = width;
    this.color = color;
}
@Override
public double area() {
    return length * width;
}
@Override
public String getColor() {
    return color;
}
}
interface Sign {
    void display();
}
class SignWithShape implements Sign {
    private Shape shape;
    private String text;
    public SignWithShape(Shape shape, String text) {
        this.shape = shape;
        this.text = text;
    }
    @Override
    public void display() {
        System.out.println("Sign Background: " + shape.getColor());
        System.out.println("Sign Text: " + text);
        System.out.println("Shape Area: " + shape.area());
    }
}
public class prac47{
    public static void main(String[] args) {
        Circle circle = new Circle(5.0, "Red");
        Rectangle rectangle = new Rectangle(4.0, 6.0, "Blue");
        SignWithShape circleSign = new SignWithShape(circle, "Circle Sign");
        SignWithShape rectangleSign = new SignWithShape(rectangle, "Rectangle Sign");
        circleSign.display();
        rectangleSign.display();
    }
}
```

OUTPUT:

```
PS D:\Probin's Work\Java Programming> javac clg.java
PS D:\Probin's Work\Java Programming> java clg
22DCE006
Sign Background: yellow
Sign Text: Circle Sign
Shape Area: 314.1592653589793
Sign Background: yellow
Sign Text: Rectangle Sign
Shape Area: 50.0
PS D:\Probin's Work\Java Programming> |
```

Staff Signature:**Grade:****Remarks by the Staff:**

PART – V Exception Handling

Practical 24

Write a java program which takes two integers x & y as input, you have to compute x/y. If x and y are not integers or if y is zero, exception will occur and you have to report it.

PROGRAM CODE:

```
import java.util.*;
public class clg{
    public static void main(String[] args) {
        System.out.println("22DCE006");
        Scanner scanner = new Scanner(System.in);
        try {
            System.out.print("Enter the numerator (integer): ");
            int x = scanner.nextInt();
            System.out.print("Enter the denominator (integer): ");
            int y = scanner.nextInt();
            int result = divide(x, y);
            System.out.println("Result: " + result);
        } catch (ArithmeticException e) {
            System.err.println("Error: Division by zero is not allowed.");
        } catch (java.util.InputMismatchException e) {
            System.err.println("Error: Please enter valid integers.");
        } finally {
            scanner.close();
        }
    }
    public static int divide(int x, int y) {
        if (y == 0) {
            throw new ArithmeticException("Division by zero");
        }
        return x / y;
    }
}
```

OUTPUT:

```
PS D:\Probin's Work\Java Programming> javac clg.java
PS D:\Probin's Work\Java Programming> java clg
22DCE006
Enter the numerator (integer): 151
Enter the denominator (integer): 0
Error: Division by zero is not allowed.
PS D:\Probin's Work\Java Programming> |
```

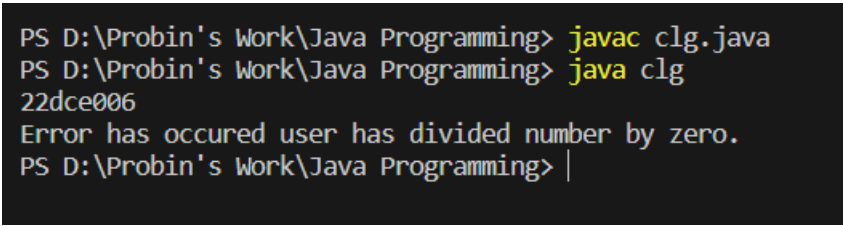
Practical 25

Write a Java program that throws an exception and catch it using a try-catch block.

PROGRAM CODE:

```
public class clg {  
    public static void main(String[] args) {  
        System.out.println("22dce006");  
        try {  
  
            int result = 10 / 0;  
            System.out.println("Result: " + result);  
        } catch (ArithmeticException e) {  
            System.err.println("Error has occurred user has divided number by zero.");  
        }  
    }  
}
```

OUTPUT:



```
PS D:\Probin's Work\Java Programming> javac clg.java  
PS D:\Probin's Work\Java Programming> java clg  
22dce006  
Error has occurred user has divided number by zero.  
PS D:\Probin's Work\Java Programming> |
```

Practical 26

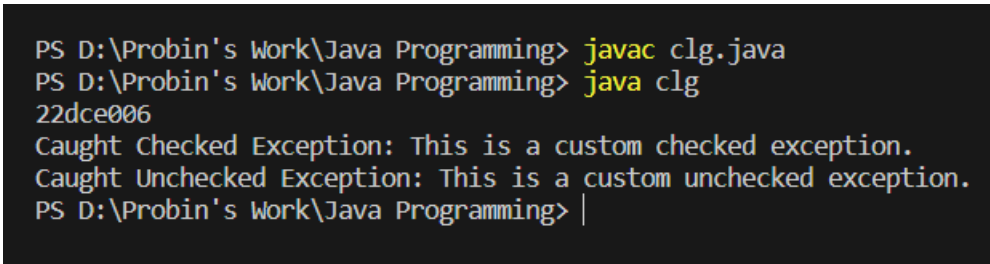
Write a java program to generate user defined exception using “throw” and “throws” keyword. Also Write a java that differentiates checked and unchecked exceptions. (Mention at least two checked and two unchecked exceptions in program).

PROGRAM CODE:

```
class CustomCheckedException extends Exception {  
    public CustomCheckedException(String message) {  
        super(message);  
    }  
}  
  
class CustomUncheckedException extends RuntimeException {  
    public CustomUncheckedException(String message) {  
        super(message);  
    }  
}
```

```
public class clg{
    public static void main(String[] args)
    {
        System.out.println("22dce006");
        try {
            throwCheckedException();
        }
        catch (CustomCheckedException e) {
            System.err.println("Caught Checked Exception: " + e.getMessage());
        }
        try {
            throwUncheckedException();
        }
        catch (CustomUncheckedException e) {
            System.err.println("Caught Unchecked Exception: " + e.getMessage());
        }
    }
    static void throwCheckedException() throws CustomCheckedException {
        throw new CustomCheckedException("This is a custom checked exception.");
    }
    static void throwUncheckedException() {
        throw new CustomUncheckedException("This is a custom unchecked exception.");
    }
}
```

OUTPUT:



```
PS D:\Probin's Work\Java Programming> javac clg.java
PS D:\Probin's Work\Java Programming> java clg
22dce006
Caught Checked Exception: This is a custom checked exception.
Caught Unchecked Exception: This is a custom unchecked exception.
PS D:\Probin's Work\Java Programming> |
```

Staff Signature:

Grade:

Remarks by the Staff:

PART-VI File Handling and Streams

Practical 27

Write a program that will count the number of lines in each file that is specified on the command line. Assume that the files are text files. Note that multiple files can be specified, as in "java Line Counts file1.txt file2.txt file3.txt". Write each file name, along with the number of lines in that file, to standard output. If an error occurs while trying to read from one of the files, you should print an error message for that file, but you should still process all the remaining files.

PROGRAM CODE:

```
import java.io.*;
public class clg {
    public static void main(String[] args) {
        System.out.println("22dce006");
        String filePath1 = "file1.txt";
        String filePath2 = "file2.txt";
        try {
            FileWriter fileWriter1 = new FileWriter(filePath1);
            BufferedWriter bufferedWriter1 = new BufferedWriter(fileWriter1);
            bufferedWriter1.write("Hello, everybody.");
            bufferedWriter1.newLine();
            bufferedWriter1.write("Written task.");
            bufferedWriter1.close();

            FileWriter fileWriter2 = new FileWriter(filePath2);
            BufferedWriter bufferedWriter2 = new BufferedWriter(fileWriter2);
            bufferedWriter2.write("Hello, everybody");
            bufferedWriter2.newLine();
            bufferedWriter2.write("Written task");
            bufferedWriter2.close();

            System.out.println("Files are done created and also written successfully.");
        }
        catch (IOException e) {
            System.out.println("An error occurred: " + e.getMessage());
        }
        for (String filePath : args) {
            try {
                FileReader fileReader = new FileReader(filePath);
                BufferedReader bufferedReader = new BufferedReader(fileReader);
                int lineCount = 0;
                while (bufferedReader.readLine() != null) {
                    lineCount++;
                }
                bufferedReader.close();
                System.out.println(filePath + ": " + lineCount + " lines");
            } catch (IOException e) {
```

```

        System.err.println("Error reading " + filePath + ": " + e.getMessage());
    }
}
}
}

```

OUTPUT:

```

PS D:\Probin's Work\Java Programming> javac clg.java
PS D:\Probin's Work\Java Programming> java clg
22dce006
Files are done created and also written successfully.
PS D:\Probin's Work\Java Programming> |

```

Practical 28

Write an example that counts the number of times a particular character, such as e, appears in a file. The character can be specified at the command line. You can use xanadu.txt as the input file.

PROGRAM CODE:

```

import java.io.*;

public class clg {
    public static void main(String[] args) {
        System.out.println("22dce006");
        createXanaduFile();
        if (args.length != 1) {
            System.out.println("Usage: java CharacterCounter <character>");
            return;
        }
        char targetCharacter = args[0].charAt(0);
        try {
            FileReader fileReader = new FileReader("xanadu.txt");
            BufferedReader bufferedReader = new BufferedReader(fileReader);
            int charCount = 0;
            int currentChar;
            while ((currentChar = bufferedReader.read()) != -1) {
                char ch = (char) currentChar;
                if (ch == targetCharacter) {
                    charCount++;
                }
            }
            bufferedReader.close();
        }
    }
}

```

```

        System.out.println("Character '" + targetCharacter + "' appears " + charCount + " times in
xanadu.txt");
    }
    catch (IOException e) {
        System.err.println("Error reading xanadu.txt: " + e.getMessage());
    }
}
private static void createXanaduFile() {
    try {
        FileWriter fileWriter = new FileWriter("xanadu.txt");
        BufferedWriter bufferedWriter = new BufferedWriter(fileWriter);
        String sampleText = "In Xanadu did Kubla Khan a stately pleasure dome decree.";
        bufferedWriter.write(sampleText);
        bufferedWriter.close();
    }
    catch (IOException e) {
        System.err.println("Error creating xanadu.txt: " + e.getMessage());
    }
}
}

```

OUTPUT:

```

PS D:\Probin's Work\Java Programming> javac clg.java
PS D:\Probin's Work\Java Programming> java clg
22dce006
Usage: java CharacterCounter <character>
PS D:\Probin's Work\Java Programming> |

```

Practical 29

Write a Java Program to Search for a given word in a File. Also show use of Wrapper Class with an example.

PROGRAM CODE:

```

import java.io.*;
import java.util.Scanner;
public class clg
{
    public static void main(String[] args) {
        System.out.println("22dce006");
        createSampleFile();
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a word ");
        String searchWord = scanner.nextLine();
        int wordCount = 0;
        try {

```

```
FileReader fileReader = new FileReader("fpp.txt");
BufferedReader bufferedReader = new BufferedReader(fileReader);
String line;
while ((line = bufferedReader.readLine()) != null) {
    String[] words = line.split(" ");
    for (String word : words) {
        String cleanedWord = word.replaceAll("[^a-zA-Z]", "").toLowerCase();
        if (cleanedWord.equals(searchWord.toLowerCase())) {
            wordCount++;
        }
    }
}
bufferedReader.close();
System.out.println("The word '" + searchWord + "' appears " + wordCount + " times");
}
catch (IOException e) {
    System.err.println("Error reading the file: " + e.getMessage());
}
}
private static void createSampleFile() {
    try {
        FileWriter fileWriter = new FileWriter("fpp.txt");
        BufferedWriter bufferedWriter = new BufferedWriter(fileWriter);
        String sampleText = "DEPSTAR CE.";
        bufferedWriter.write(sampleText);
        bufferedWriter.close();
    }
    catch (IOException e) {
        System.err.println("Error creating the fpp.txt file: " + e.getMessage());
    }
}
}
```

OUTPUT:

```
PS D:\Probin's Work\Java Programming> javac clg.java
PS D:\Probin's Work\Java Programming> java clg
22dce006
Enter a word hello
The word 'hello' appears 0 times
PS D:\Probin's Work\Java Programming> |
```

Practical 30

Write a program to copy data from one file to another file. If the destination file does not exist, it is created automatically.

PROGRAM CODE:

```
import java.util.*;
import java.io.*;

public class clg{
    public static void main(String[] args) throws IOException,FileNotFoundException {
        System.out.println("22dce006");
        String source, destination;
        FileReader source_f;
        File f;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Filename: ");
        source = sc.nextLine();
        source_f = new FileReader(source);
        System.out.println("Enter Destination Filename : ");
        destination = sc.nextLine();
        f = new File(destination);
        FileWriter destination_f;
        if(!f.exists())
            f.createNewFile();
        destination_f = new FileWriter(destination);
        int c = source_f.read();
        while(c!=-1) {
            destination_f.write(c);
            c = source_f.read();
        }
        System.out.println("File Copy operation done");
        source_f.close();
        destination_f.close();
        sc.close();
    }
}
```


OUTPUT:

```
PS D:\Probin's Work\Java Programming> javac clg.java
PS D:\Probin's Work\Java Programming> java clg
22dce006
Enter Filename:
fpp.txt
Enter Destination Filename :
op.txt
File Copy operation done
PS D:\Probin's Work\Java Programming> |
```

Practical 31

Write a program to show use of character and byte stream. Also show use of `BufferedReader/BufferedWriter` to read console input and write them into a file.

PROGRAM CODE:

```
import java.io.*;
public class clg{
    public static void main(String[] args) {
        System.out.println("22dce006");
        try{
            File f1 = new File("fpp.txt");
            FileWriter fw = new FileWriter(f1);
            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
            BufferedWriter bw = new BufferedWriter(fw);
            System.out.println("Enter input: ");
            String s = new String(br.readLine());
            fw.write(s,0,s.length());
            fw.close();
            bw.close();
            br.close();
            System.out.println("Data Write operation done");
        }
        catch(IOException e){
            System.out.println("error");
            e.printStackTrace();
        }
    }
}
```

OUTPUT:

```
PS D:\Probin's Work\Java Programming> javac clg.java
PS D:\Probin's Work\Java Programming> java clg
22dce006
Enter input:
hello world
Data Write operation done
PS D:\Probin's Work\Java Programming> |
```

Staff Signature:**Grade:****Remarks by the Staff:**

PART-VII Multithreading

Practical 32


Write a program to create thread which display “Hello World” message. A. by extending Thread class B. by using Runnable interface.

PROGRAM CODE:

```
class HelloWorldRunnable implements Runnable {
    @Override
    public void run() {
        System.out.println("Hello World prac 32");
    }
}

public class clg {
    public static void main(String[] args) {
        System.out.println("22dce006");
        HelloWorldRunnable helloWorldRunnable = new HelloWorldRunnable();
        Thread thread = new Thread(helloWorldRunnable);
        thread.start();
    }
}
```

OUTPUT:



```
PS D:\Probin's Work\Java Programming> javac clg.java
PS D:\Probin's Work\Java Programming> java clg
22dce006
Hello World prac 32
PS D:\Probin's Work\Java Programming> |
```

Practical 33

Write a java program which takes N and number of threads as an argument. Program should distribute the task of summation of N numbers amongst number of threads and final result to be displayed on the console.

PROGRAM CODE:

```
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.atomic.AtomicLong;
```

```
class SumTask implements Runnable {
    private long start;
    private long end;
    private AtomicLong result;

    public SumTask(long start, long end, AtomicLong result) {
        this.start = start;
        this.end = end;
        this.result = result;
    }

    @Override
    public void run() {
        long localSum = 0;
        for (long i = start; i <= end; i++) {
            localSum += i;
        }
        result.addAndGet(localSum);
    }
}

public class clg {
    public static void main(String[] args) {
        System.out.println("22dce006");
        if (args.length != 2) {
            System.err.println("JAVA SUM PARALLEL <numThreads>");
            System.exit(1);
        }
        long N = Long.parseLong(args[0]);
        int numThreads = Integer.parseInt(args[1]);

        AtomicLong result = new AtomicLong(0);
        ExecutorService executorService = Executors.newFixedThreadPool(numThreads);

        long chunkSize = (N + numThreads - 1) / numThreads;
        for (int i = 0; i < numThreads; i++) {
            long start = i * chunkSize + 1;
            long end = Math.min((i + 1) * chunkSize, N);
            executorService.execute(new SumTask(start, end, result));
        }
        executorService.shutdown();
        while (!executorService.isTerminated()) {

        }
        System.out.println("Sum of numbers from 1 to " + N + " is: " + result.get());
    }
}
```

OUTPUT:

```
PS D:\Probin's Work\Java Programming> javac clg.java
PS D:\Probin's Work\Java Programming> java clg
22dce006
JAVA SUM PARALLEL <numThreads>
PS D:\Probin's Work\Java Programming> |
```

Practical 34

Write a java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.

PROGRAM CODE:

```
import java.util.Random;
class NumberProcessor extends Thread {
    private int number;

    public NumberProcessor(int number) {
        this.number = number;
    }
    @Override
    public void run() {
        if (number % 2 == 0) {
            int square = number * number;
            System.out.println("Generated Number: " + number);
            System.out.println("Square: " + square);
        } else {
            int cube = number * number * number;
            System.out.println("Generated Number: " + number);
            System.out.println("Cube: " + cube);
        }
    }
}

public class clg {
    public static void main(String[] args) {
        System.out.println("22dce006");
        Random random = new Random();

        for (int i = 0; i < 5; i++) {
            int randomNumber = random.nextInt(100);
            NumberProcessor numberProcessor = new NumberProcessor(randomNumber);
            numberProcessor.start();
        }
    }
}
```

```
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
```

OUTPUT:

```
PS D:\Probin's Work\Java Programming> javac clg.java
PS D:\Probin's Work\Java Programming> java clg
22dce006
Generated Number: 53
Cube: 148877
Generated Number: 18
Square: 324
Generated Number: 76
Square: 5776
Generated Number: 48
Square: 2304
Generated Number: 70
Square: 4900
PS D:\Probin's Work\Java Programming> |
```

Practical 35

Write a java program to increment the value of one variable by one and display it after one second using thread using sleep() method.

PROGRAM CODE:

```
class IncrementThread extends Thread {
    private int value;
    public IncrementThread(int value) {
        this.value = value;
    }
    @Override
    public void run() {
        try {
            Thread.sleep(100);
        }
        catch (InterruptedException e) {
            e.printStackTrace();
        }
        value++;
        System.out.println("After increment: " + value);
    }
}
```

```

public class clg{
    public static void main(String[] args) {
        System.out.println("22dce006");
        int initialValue = 17;
        IncrementThread incrementThread = new IncrementThread(initialValue);
        incrementThread.start();
    }
}

```

OUTPUT:

```

PS D:\Probin's Work\Java Programming> javac clg.java
PS D:\Probin's Work\Java Programming> java clg
22dce006
After increment: 18
PS D:\Probin's Work\Java Programming> |

```

Practical 36

Write a java program to create three threads ‘FIRST’, ‘SECOND’, ‘THIRD’. Set the priority of the ‘FIRST’ thread to 3, the ‘SECOND’ thread to 5(default) and the ‘THIRD’ thread to 7.

PROGRAM CODE:

```

public class clg {
    public static void main(String[] args) {
        System.out.println("22dce006");

        Thread firstThread = new Thread(new MyRunnable(), "FIRST");
        firstThread.setPriority(3);

        Thread secondThread = new Thread(new MyRunnable(), "SECOND");

        Thread thirdThread = new Thread(new MyRunnable(), "THIRD");
        thirdThread.setPriority(7);

        firstThread.start();
        secondThread.start();
        thirdThread.start();
    }
    static class MyRunnable implements Runnable {
        @Override
        public void run() {
            for (int i = 1; i <= 5; i++) {
                System.out.println(Thread.currentThread().getName() + " = Count: " + i);
                try {

```

```

        Thread.sleep(100);
    }
    catch (InterruptedException e) {
        e.printStackTrace();
    }
}
}
}
}
}

```

OUTPUT:

```

PS D:\Probin's Work\Java Programming> javac clg.java
PS D:\Probin's Work\Java Programming> java clg
22dce006
THIRD = Count: 1
FIRST = Count: 1
SECOND = Count: 1
FIRST = Count: 2
THIRD = Count: 2
SECOND = Count: 2
FIRST = Count: 3
THIRD = Count: 3
SECOND = Count: 3
SECOND = Count: 4
FIRST = Count: 4
THIRD = Count: 4
THIRD = Count: 5
FIRST = Count: 5
SECOND = Count: 5
PS D:\Probin's Work\Java Programming> |

```

Practical 37

Write a java program to solve producer-consumer problem using thread synchronization.

PROGRAM CODE:

```

import java.util.LinkedList;
class ProducerConsumer {
    LinkedList<Integer> buffer = new LinkedList<>();
    int capacity = 5;

    public void produce() throws InterruptedException {
        for (int i = 0; i < 4; i++) {
            int item = i + 1;
            synchronized (this) {

```



```
        while (buffer.size() == capacity) {
            wait();
        }
        System.out.println("Producer=" + item);
        buffer.add(item);
        notify();
        Thread.sleep(1000);
    }
}
}
public void consume() throws InterruptedException {
    for (int i = 0; i < 4; i++) {
        synchronized (this) {
            while (buffer.isEmpty()) {
                wait();
            }
            int item = buffer.poll();
            System.out.println("Consumer= " + item);
            notify();
            Thread.sleep(1000);
        }
    }
}
}

public class clg{
    public static void main(String[] args) {
        System.out.println("22dce006");
        ProducerConsumer pc = new ProducerConsumer();
        Thread producerThread = new Thread(() -> {
            try {
                pc.produce();
            }
            catch (InterruptedException e) {
                e.printStackTrace();
            }
        });
        Thread consumerThread = new Thread(() -> {
            try {
                pc.consume();
            }
            catch (InterruptedException e) {
                e.printStackTrace();
            }
        });
        producerThread.start();
        consumerThread.start();
    }
}
```

OUTPUT:

```
PS D:\Probin's Work\Java Programming> javac clg.java
PS D:\Probin's Work\Java Programming> java clg
22dce006
Producer=1
Producer=2
Producer=3
Producer=4
Consumer= 1
Consumer= 2
Consumer= 3
Consumer= 4
PS D:\Probin's Work\Java Programming> |
```

Staff Signature:**Grade:****Remarks by the Staff:**

PART-VIII Collection Framework and Generic

Practical 38

Design a Custom Stack using ArrayList class, which implements following functionalities of stack. My Stack

-list ArrayList<Object>: A list to store elements.

+isEmpty: boolean: Returns true if this stack is empty.

+getSize(): int: Returns number of elements in this stack.

+peek(): Object: Returns top element in this stack without removing it.

+pop(): Object: Returns and Removes the top elements in this stack.

+push(o: object): Adds new element to the top of this stack.

PROGRAM CODE:

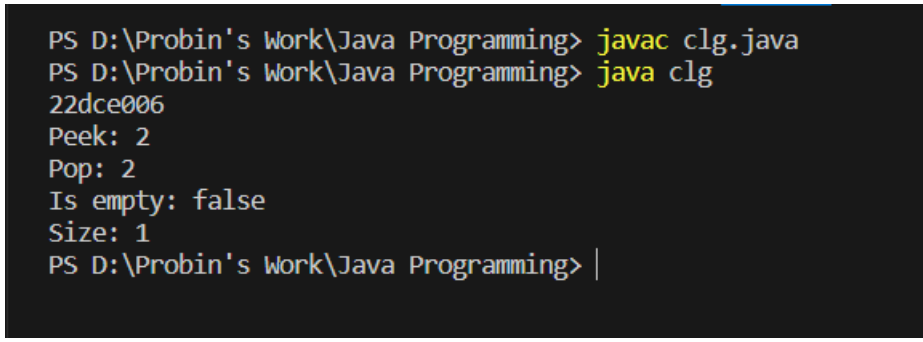
```
_import java.util.ArrayList;

class CustomStack {
    private ArrayList<Object> list = new ArrayList<>();
    public boolean isEmpty() {
        return list.isEmpty();
    }
    public int getSize() {
        return list.size();
    }
    public Object peek() {
        if (!isEmpty()) {
            return list.get(list.size() - 1);
        }
        else {
            throw new IllegalStateException("Stack is empty");
        }
    }
    public Object pop() {
        if (!isEmpty()) {
            return list.remove(list.size() - 1);
        }
        else {
            throw new IllegalStateException("Stack is empty");
        }
    }
    public void push(Object o) {
        list.add(o);
    }
}

public class clg{
    public static void main(String[] args) {
        System.out.println("22dce006");
        CustomStack stack = new CustomStack
```

```
stack.push(1);
stack.push(2);
System.out.println("Peek: " + stack.peek);
System.out.println("Pop: " + stack.pop());
System.out.println("Is empty: " + stack.isEmpty);
System.out.println("Size: " + stack.getSize());
}
}
```

OUTPUT:



```
PS D:\Probin's Work\Java Programming> javac clg.java
PS D:\Probin's Work\Java Programming> java clg
22dce006
Peek: 2
Pop: 2
Is empty: false
Size: 1
PS D:\Probin's Work\Java Programming> |
```

Practical 39

Create a generic method for sorting an array of Comparable objects.

PROGRAM CODE:

```
import java.util.*;
public class clg{
    public static void sort(ArrayList<Integer> arr){
        Collections.sort(arr);
    }
    public static void main(String[] args){
        System.out.println("22dce006");
        ArrayList<Integer> arr= new ArrayList<Integer>();
        for(int i =5; i < 11; ++i){
            arr.add(i);
        }
        sort(arr);
        for(int i=0; i < arr.size(); ++i){
            System.out.println(arr.get(i));
        }
    }
}
```

OUTPUT:

```
PS D:\Probin's Work\Java Programming> javac clg.java
PS D:\Probin's Work\Java Programming> java clg
22dce006
5
6
7
8
9
10
PS D:\Probin's Work\Java Programming> |
```

Practical 40

Write a program that counts the occurrences of words in a text and displays the words and their occurrences in alphabetical order of the words. Using Map and Set Classes.

PROGRAM CODE:

```
import java.util.*;
public class clg{
    public static void main(String[] args)
    {
        System.out.println("22dce006");
        Scanner sc = new Scanner(System.in);
        System.out.print("Input your text : ");
        String s = sc.nextLine();
        s = s.toLowerCase();
        StringTokenizer st = new StringTokenizer(s,".,/()! ");
        TreeSet<String> words = new TreeSet<String>();
        while(st.hasMoreTokens()){
            words.add( st.nextToken() );
        }
        HashMap<String, Integer> map = new HashMap<String, Integer>();
        Iterator<String> i = words.iterator();
        while(i.hasNext()){
            String word = i.next();
            int count = 0;
            st = new StringTokenizer(s,".,/()!");
            while(st.hasMoreTokens()){
                String swe = st.nextToken();
                if(word.equals(swe) ) count++;
            }
            map.put( word, count );
        }
        System.out.println(map);
    }
}
```

```
        sc.close();  
    }  
}
```

OUTPUT:

```
PS D:\Probin's Work\Java Programming> javac clg.java  
PS D:\Probin's Work\Java Programming> java clg  
22dce006  
Input your text : depstar  
{depstar=1}  
PS D:\Probin's Work\Java Programming> |
```

Practical 41

Write a code which counts the number of the keywords in a Java source file. Store all the keywords in a HashSet and use the contains () method to test if a word is in the keyword set.

PROGRAM CODE:

```
import java.util.*;
import java.io.*;
public class clg{
    public static void main(String[] args) throws IOException {
        System.out.println("22dce006");

        String k[] = new String[] {"abstract", "assert",
        "boolean", "break", "byte", "case", "catch", "char", "class",

        "continue", "default", "do", "double", "else", "enum ", "extends", "final", "finally",

        "float", "for", "if", "implements", "import", "instanceof", "int", "interface", "long",
        "native", "new", "package", "private", "protected", "public", "return", "short", "c",
        "strictfp", "super", "switch", "synchronized", "this", "throw", "throws", "transient", "try",
        "void", "volatile", "while"};

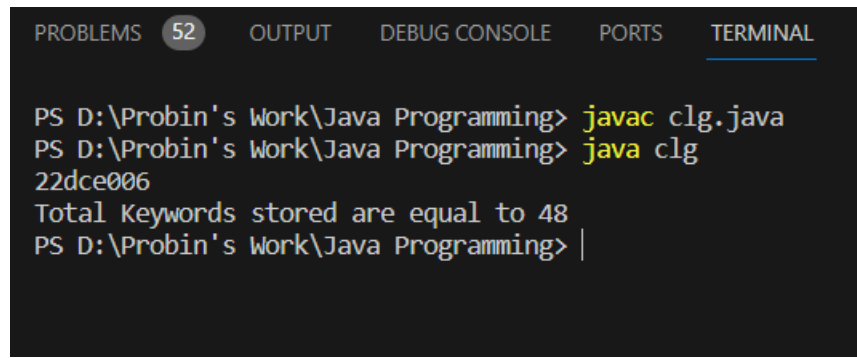
        HashSet<String> s = new HashSet<String>();

        for(int i =0;i < k.length; ++i)
        {
            s.add( k[i] );
        }

        int count = 0;
        Iterator<String> it = s.iterator();
        while (it.hasNext()) {
            count++;
            it.next();
        }

        System.out.println("Total Keywords stored are equal to " +count);

    }
}
```

OUTPUT:

The screenshot shows an IDE interface with a terminal window. The terminal has tabs for PROBLEMS (52), OUTPUT, DEBUG CONSOLE, PORTS, and TERMINAL. The terminal output shows the following commands and results:

```
PS D:\Probin's Work\Java Programming> javac clg.java
PS D:\Probin's Work\Java Programming> java clg
22dce006
Total Keywords stored are equal to 48
PS D:\Probin's Work\Java Programming> |
```

Staff Signature:**Grade:****Remarks by the Staff:**